CLAIMS

What is claimed is:

1. A caching method, comprising:

caching first data received from a data source within a static cache as stable data, the static cache having a fixed size;

evicting portions of the stable data within the static cache to a dynamic cache when the static cache is full; and

enrolling the evicted portions of the stable data into the dynamic cache as soft data, the dynamic cache having a dynamic size.

2. The caching method of claim 1, wherein the dynamic cache is dynamically sized according to availability of memory.

3. The caching method of claim 2, wherein evicting the portions of the stable data further comprises evicting the portions of the stable data to the dynamic cache according to a Least Recently Used eviction policy.

4. The caching method of claim 2, further comprising:

evicting selectively at least some of the soft data from the dynamic cache when the availability of the memory is scarce; and

contracting the dynamic cache to release some of the memory consumed by the dynamic cache.

5. The caching method of claim 4, wherein evicting selectively the at least some of the soft data further comprises evicting the at least some of the soft data according to a Least Recently Used eviction policy.

6. The caching method of claim 4, wherein enrolling the evicted portions of the stable data into the dynamic cache as soft data comprises caching the soft data as hash values of a hash table, the hash values being indexed to keys for accessing the hash values.

7. The caching method of claim 6, wherein evicting selectively at least some of the soft data from the dynamic cache comprises:

copying at least some of the keys into a garbage queue, the at least some of the keys corresponding to the at least some of the soft data; and

removing at least some of the hash values from the hash table based on the at least some of the keys in the garbage queue.

8. The caching method of claim 7, wherein a Java Garbage Collector selectively copies the at least some of the keys into the garbage queue.

9. The caching method of claim 2, further comprising:

intercepting a request for second data from the data source;

determining whether the second data is cached within either of the static cache and dynamic cache; and

providing the second data from either of the static cache and the dynamic cache instead of the data source, if the determining determines that the second data is cached.

10. The caching method of claim 9, further comprising moving the second data to a most recently used position within the static cache, if the determining determines that the second data is cached.

11. The caching method of claim 2, wherein the static cache and the dynamic cache comprise a hybrid-cache within a single memory device.

12. The caching method of claim 2, wherein the stable data and the soft data comprise objects of an object orientated language.

13. A machine-accessible medium that provides instructions that, if executed by a machine, will cause the machine to perform operations comprising:

caching first data received from a data source into a hybrid-cache, the hybrid-cache including a static cache having a fixed size and a dynamic cache having a dynamic size;

enrolling the first data received from a data source into the static cache as stable data;

evicting selective portions of the stable data within the static cache to the dynamic cache when the static cache is full; and

enrolling the selective portions of the stable data evicted from the static cache into the dynamic cache as soft data.

14. The machine-accessible medium of claim 13, wherein the dynamic cache is dynamically sized according to availability of memory.

15. The machine-accessible medium of claim 14, further providing instructions that, if executed by the machine, will cause the machine to perform further operations, comprising:

expanding the dynamic cache to accommodate the selective portions of the stable data evicted to the dynamic cache, if adequate memory is available; and

evicting at least some of the soft data from the dynamic cache to accommodate the selective portions of the stable data evicted to the dynamic cache, if adequate memory is not available.

16. The machine-accessible medium of claim 15, further providing instructions that, if executed by the machine, will cause the machine to perform further operations, comprising:

contracting the dynamic cache to release some of the memory consumed by the dynamic cache, if the memory is scarce.

17. The machine-accessible medium of claim 15, wherein enrolling the selective portions of the stable data evicted from the static cache into the dynamic cache as the soft data comprises caching the soft data within the dynamic cache according to a canonical mapping scheme.

18. The machine-accessible medium of claim 17, wherein caching the soft data within the dynamic cache according to the canonical mapping scheme comprises caching the soft data as a hash value of a hash table, the hash values being indexed to keys for accessing the hash values.

19. The machine-accessible medium of claim 18, wherein evicting the at least some of the soft data from the dynamic cache comprises:

copying at least some of the keys into a garbage queue, the at least some of the keys corresponding to the at least some of the soft data; and

removing at least some of the hash values from the hash table based on the at least some of the keys in the garbage queue.

20. The machine-accessible medium of claim 13, wherein evicting selective portions of the stable data within the static cache comprises evicting the selective portions of the stable data according to a Least Recently Used eviction policy.

21. The machine-accessible medium of claim 13, wherein the stable data and the soft data comprise objects of an object orientated language.

22. A system, comprising:

a processor to process requests for first data from a data source; and

a memory device communicatively coupled to the processor, the memory device to hold a hybrid-cache, the hybrid-cache comprising:

a static cache for caching the first data as stable data, the static cache having a fixed size; and

a dynamic cache having a dynamic size according to availability of memory within the memory device, wherein portions of the stable data within the static cache are to be evicted to the dynamic cache as soft data when the static cache is full.

23. The system of claim 22, wherein the dynamic cache is to expand to accommodate the portions of the stable data evicted to the dynamic cache when the static cache is full, if adequate memory is available within the memory device.

24. The system of claim 23, wherein the dynamic cache is further to evict at least some of the soft data from the dynamic cache to accommodate the portions of the stable data evicted to the dynamic cache, if adequate memory is not available within the memory device.

25. The system of claim 24, wherein the dynamic cache is further to contract to release memory consumed by the dynamic cache, if other entities within the memory device expand.

26. The system of claim 24, wherein the memory device comprises Random Access Memory ("RAM") and wherein the data source comprises a data storage device communicatively coupled to the processor, the hybrid-cache to reduce swapping to the data storage device.

27. The system of claim 22, wherein the system comprises a caching server, wherein the requests for the first data from the data source comprise requests from clients of the caching server, and wherein the data source comprises an Internet.

28. The system of claim 22, wherein the system comprises an Application Server, wherein the requests for the first data from the data source comprise requests from clients of the Application Server, and wherein the data source comprises at least one database.

29. The system of claim 22, wherein the Application Server comprises one of a Java based Application Server and a .NET based Application Server.

30. A system, comprising:

static means for caching stable data received from a data source within a fixed amount of memory;

first means for selectively evicting portions of the stable data from the static means when the static means is full;

dynamic means for caching soft data within a dynamically changing amount of memory; and

means for enrolling the portions of the stable data evicted by the means for evicting into the dynamic means as the soft data.

31. The system of claim 30, wherein the dynamic means is further for caching the soft data within the dynamically changing amount of the memory based on an available amount of the memory.

32. The system of claim 31, further comprising:

second means for evicting the soft data from the dynamic means when the available amount of memory is scarce.

33. The system of claim 33, wherein the dynamic means is further for contracting the dynamically changing amount of memory when the available amount of memory is scarce.